



Request DELETE

Transcripción

[00:00] Bien, ya tenemos el último requerimiento para terminar nuestro CRUD: el delete, y eso ya es la exclusión de médicos o deletes de médicos, como quieran llamarlo.

[00:12] Ahora, en las reglas negocio aquí hay algo muy curioso y es que dice: El registro no debe ser borrado de la base de datos. Y el listado debe retornar solo médicos activos. Esto ya me está diciendo que el delete que yo debo hacer es un delete lógico.

[00:29] Es una exclusión lógica y no una exclusión, por así decirlo física, o sea, lo que yo quiero es que el médico quede desactivado a nivel de base de datos, pero yo no quiero que esa línea se elimine de mi base de datos porque obviamente siempre es bueno mantener un histórico, qué médicos han trabajado aquí.

[00:50] Por ejemplo venimos aquí y supongamos pues que Diego López, mi médico, que está aquí, él ya renunció al trabajo, entonces yo tengo la opción de borrar este registro de la base de datos, con lo cual yo ya no tengo ninguna prueba histórica de que Diego López alguna vez hizo parte de voll.medClinic o puedo simplemente desactivar su registro, que es un delete lógico, exclusión lógica.

[01:16] Vamos por partes. Lo primero, como siempre, voy a crear mi request. Voy a decirle aquí que este request se va a llamar delete médico. Voy a renombrar y le digo Delete médico. Hago un renombre y voy a copiar la misr

URL porque ya vimos que es igual. Pero el verbo HTTP va a ser un delete como hemos visto aquí.

[01:46] Entonces ya tengo mi método que va a borrar médico, si lo ejecuto nuevamente método no permitido porque no está implementado. Ahora, el delete tiene algo muy curioso, y es que al igual que yo necesito saber el id por ejemplo, al igual yo tengo que venir acá y traer al médico por el id, el médico que yo quiero actualizar, delete también necesita saber que médico es el que va a eliminar.

[02:16] Vamos a dividir esto en dos partes, primero vamos a hacer el delete común, el delete verdadero. Vamos a borrar un médico de la base de datos y después ya lo damos a modificar para que haga la exclusión lógica. ¿Cómo le decimos ahora al método delete “bórrame el médico con el id 3”, por ejemplo? En el listado tenemos el id 3.

[02:39] ¿Cómo le digo, “quiero que el médico con id 3 sea eliminado?” Bueno, se lo puedo enviar en el body, puedo incluir aquí un parámetro, puedo crear un DTO con un solo parámetro, que diga id y mandárselo, algo como eso de repente. Puedo crear aquí un JSON con id, igual y el id que yo quiero enviar. Pero no es la idea, no es la idea.

[03:05] Aquí yo tengo una forma un poco más interesante de hacerlo y es usando path variables. ¿Qué es un path variable? Path variable es una variable que va en la URL, en el path, este de aquí es el path. Entonces un path variable es una variable que va aquí, por ejemplo el id que yo quería eliminar era el id 3.

[03:28] Entonces sería localhost:8080/médicos/3. Si yo le doy igual a enviar, me va a decir not found. ¿Por qué? Porque esta URL de aquí no existe. Cuando la UR no existe, ¿qué es lo que hacemos? Crearla. Entonces, igual que antes vengo aquí y le voy a decir public void eliminarMedico. Abro paréntesis, y si ya tengo, por ejemplo, putMapping, postMapping, getMapping. ¿Qué creen que puede ir aquí? DeleteMapping.

[04:02] Entonces vamos a `@DeleteMapping`, perfecto, eliminarMedico. Y vamos a dejarlo ahí por ahora porque ya se supone que mi delete ya está mapeado. Ahora voy a guardar. Vamos a limpiar nuestra terminal. Esperamos que recargue mi servidor. Vemos que recargó. Envío y vemos que igual no encuentra la URL, porque el slash 3 no está mapeado.

[04:38] Nosotros tenemos el `@RequestMapping` hasta médicos, ¿qué podemos hacer aquí? Simple, voy a decirle que en el `@DeleteMapping`, abre los paréntesis y aquí le voy a decir, aquí no, tiene que ser dentro de los paréntesis, y le voy a decir (`"/3"`). Con eso voy a mapear mi URL correctamente. Espero que cargue mi servidor nuevamente.

[05:01] Vemos que recargó. Voy a darle enviar y me va a dar okay, encontró esta URL, mi URL ya existe, ¿pero qué pasaría si yo no quiero eliminar el id 3? Yo quiero eliminar el id 5, 10, 1. Esto no puede ser fijo, esto tiene que ser dinámico. ¿Y cómo me aseguro que sea dinámico? Bueno. Abrimos llaves y le vamos a dar un nombre a esta variable, a este path variable que va a ser id.

[05:33] Con esto Spring ya va identificar que aquí va a ir una variable. A este nivel, no sabe qué va a ir, pero va a ir a alguna variable. Va a ir un id aquí. Si guardo aquí, por ejemplo vamos a esperar a que cargue mi servidor nuevamente. Recargó, voy a enviar y me da 200 a pesar de que ya no lo he mapeado con `/3`. ¿Por qué? Porque ya esta variable es dinámica. Primer paso ejecutado.

[06:04] Segundo paso, obtener esta variable para trabajar con ella. ¿Qué es lo que tengo que hacer? Bueno, simplemente tengo que declarar mi variable, entonces le voy a decir aquí (`Long id`). Pero si recuerdan, Spring mágicamente no detecta de dónde viene esta variable.

[06:22] Puede venir del body, puede venir de un query param o puede venir de aquí de un path param. Como le vamos a indicar a Spring, por si acaso, esto viene de un path param. de un parámetro en el path, vamos a decirle que es un

path variable. Entonces con esto Spring lo que va a hacer es “viene del path variable id”.

[06:44] Entonces automáticamente el valor que yo le asigne aquí lo va a asignar a mi variable local, aquí en mi servidor, con el tipo de dato long. Segundo, al igual que hicimos aquí, vamos a obtener nuestro médico que queremos eliminar. Este de aquí. Pero en lugar de mandarle el DTO, vamos directamente decirle: “agarra el id que te hemos enviado”.

[07:14] Entonces borramos esto. A ver que ocurre un desastre y aquí solamente id. Listo. Entonces ahora el médico va a ser `medicoRepository.getReferenceById`, y con esto él ya va a eliminar el médico. Ahora vamos a `medicoRepository.delete` y le vamos a mandar el médico como entidad que queremos borrar.

[07:37] Y como aprendimos aquí, si queremos hacer una transacción en la base de datos, si queremos que nuestros cambios sean commiteados, vamos a agregarle pues un `@Transactional` aquí también para que tenga tengan efecto estos cambios a nivel de base de datos.

[07:56] Vemos que ya recargó y vamos a probar nuestro método, vamos a listar nuestros médicos primero. Vemos que el id 3 existe, voy a hacerle un delete y voy a borrar el médico con el id 3, vamos a ver qué me dice. 200 OK. Voy a mi listado, lo envío y ya no tengo más ese médico con el id 3, ese médico ya se eliminó por completo de mi base de datos.

[08:24] Pero por reglas del negocio, el registro no debe ser borrado de la base de datos. Hasta ahora ya sabemos cómo funciona el delete a nivel de base de datos. Ya hemos eliminado el médico de la base de datos. Pero no es lo que queremos por ahora, en el siguiente video vamos a actualizar este método para hacer un delete lógico, nos vemos.

